Available online at www.sciencedirect.com

ScienceDirect

Energy Procedia 142 (2017) 3069–3074

Energy Procedia

www.elsevier.com/locate/procedia

9th International Conference on Applied Energy, ICAE2017, 21-24 August 2017, Cardiff, UK

Scalable Local Energy Management Systems

Amir Javed[a]*, Omer F. Rana[a], Liana M. Cipcigan[b], Charalampos Marmaras[b]

[a]*School of Computer Science and Informatics, Cardiff University, Queen's Buildings, The Parade, Cardiff, CF24 3AA*
[b]*School of Engineering Cardiff University, Queen's Buildings, The Parade, Cardiff, CF24 3AA*

Abstract

Commercial buildings have been identified as a major contributor of total global energy consumption. Mechanisms for collecting data about energy consumption patterns within buildings, and their subsequent analysis to support demand estimation (and reduction) remain important research challenges, which have already attracted considerable work. We propose a cloud based energy management system that enables such analysis to scale to both increasing data volumes and number of buildings. We consider both energy consumption and storage to support: (i) flattening the peak demand of commercial building(s); (ii) enable a “cost reduction” mode where the demand of a commercial building is reduced for those hours when a “triad peak” is expected; and (iii) enables a building manager to participate in grid balancing services market by means of demand response. The energy management system is deployed on a cloud infrastructure that adapts the number of computational resources needed to estimate potential demand, and to adaptively run multiple what-if scenarios to choose the most optimum configuration to reduce building energy demand.

Peer-review under responsibility of the scientific committee of the 9th International Conference on Applied Energy.

Keywords: Electric vechiles , Energy Storage, Energy Management System, Cloud Computing

1. Introduction

Existing building energy management systems have attempted to address the need for optimal balancing of electricity demand between the customer and the energy provider(s). They utilise renewable energy resources such as wind and photovoltaic/solar, and energy storage capability (e.g. lead-acid batteries) to manage supply and demand.

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: A.Javed@cs.cardiff.ac.uk

1876-6102
Peer-review under responsibility of the scientific committee of the 9th International Conference on Applied Energy .
10.1016/j.egypro.2017.12.446

With the use of distributed systems and data capture technologies (e.g cloud and sensor-based systems), a more efficient and adaptive smart grid is now becoming available, enabling the capture and processing of real time data, with support for cybersecurity and privacy/anonymization of customer data.

Researchers have proposed distributed energy management systems to overcome challenges arising from increasing types of generation and storage sources, the ability to incorporate both consumers/produces (often referred to as "prosumers") in a peer-to-peer manner, and the ability to support varying types of loads, such as electric vehicles. To illustrate, fossil fuel based generators were used by customers to tackle peaks in demand[1][2], mechanisms to limited susceptibility due to distributed denial of service attacks[3], a single point of failure, and the ability to support a limited number of customers – all due to the use of a centralised systems architecture. However, energy management systems built on this architecture also had constraints due to limited memory and computational power[4], [5]. This constraint limited scalability (in number of customers served) and the inability to re-schedule loads and shift demand.

To address these challenges various cloud based energy management systems have been proposed, e.g. Kim et al. [3] proposed a model that focused on reducing response time for large scale deployments. Their proposed model had two aspects, one looking at data centric communication and the other looking at topic based interactions with customers. Similarly, [2] proposed a model to address peak demand with dynamic pricing requirements. Their proposed model was deployed on a cloud infrastructure so that requests from customers were executed only when computational resources became available and scheduled based on the priority of requests. Their approach capitalized on the adaptive nature of a cloud-based system, and featured dynamic allocation of bandwidth and the ability to execute jobs in parallel so all customer were served. Rajeev and Ashok [6] proposed a model that used virtualization to enable integration of existing energy storage devices. Simmhan et al. [7] provide support for demand response as Software-as-a-Service for smart grids. Using this approach, a user is able to submit data to such an externally hosted service to enable demand response analysis to be carried out. Their model intelligently managed load to reduce peak demand. Tang et al. [8] proposed a cloud hosted agent model in which agents acted as an intermediary between various components used to realise smart grid functionality to fulfil user requests. Recent work has focused on integrating fog computing paradigm with an energy management system to reduce computational load on the cloud system, to make more effective use of locally available computational resources and to enable fault tolerance due to intermittent data network connectivity and availability [9].

1.1 Contribution

Using traditional approach a local energy management system was designed that was capable of forecasting building demand based on weather attributes and was able to forecast most probable energy triad hours occurring in Great Britain [10]. By identify the most influential weather attributes for each commercial building, researchers developed a local energy management system that is capable of forecasting building energy demand, and forecasted the most probable triad hour [11]. Additionally, the model can be used to reduce peak demand of a building during triad hours by using electric vehicles and energy storage units. The cloud-based approach enables various scenarios to be executed concurrently, leading to the generation of an ensemble of models, where each model differs based on assumptions and initial conditions (such as number of expected electric vehicles at any particular time, their arrival/departure times, their likely state of charge on departure, influence of weather attributes, etc.) We show how a building manager can, at any point of time, update the properties of the system being simulated on the cloud system. The availability of computational resources that can be acquired (and released) in an elastic manner remains an important benefit of using a cloud-based approach, enabling the system to adapt based on variation in data size and user requirements.

2. Local Energy Management architecture on comet cloud.

We have extended the local energy management system[11] to reduce the total demand of a commercial building. In the proposed model specialist units are used to store energy at periods where generation exceeds demand, and to discharge this into the building at other times (e.g. during periods of high tariff). The proposed model can be operated in three modes:

<u>Peak shaving mode</u>: the aim is to flatten the aggregated demand profile of a commercial building by creating a schedule that focuses on discharging energy from electric vehicles and energy storage units at peak hours and charging these devices at non-peak hours. Hence the load is shifted from the peak hour and the resulting load profile is flattened.

Cost reduction mode: the aim is to forecast most probable energy triad hours [10], [11] and (likely) demand during these hours is significant in determining total energy cost for a commercial building.

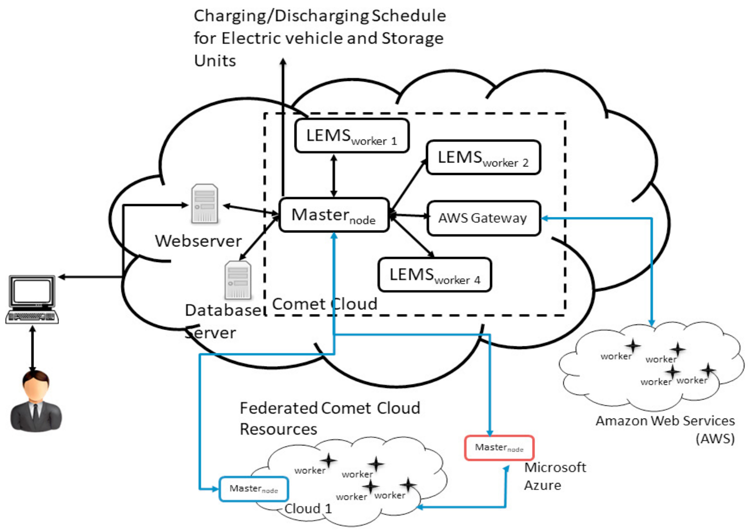

Figure 1 Architecture of energy management system deployed using CometCloud

Demand response mode: enables a building manager to increase or decrease the total demand of a commercial building. By doing so a building manager can participate in ancillary service market. Figure 1 illustrates the architecture of the deployment of local energy management systems using the CometCloud[12] framework. The main components include: (i) a webserver that hosts the graphical user interface through which the building manager can operate the local energy management system; (ii) a database server that stores relevant information about the electric vehicles, energy storage units and power consumption for each building, (iii) the CometCloud master/worker system that executes the data analysis algorithm(s).

CometCloud has been designed specifically to enable integration of multiple computational resources (through a process called “cloud federation”). Using this approach, computational resources hosted locally (e.g. a local server or an entire private cloud system) can be combined with remotely accessed services (e.g. from Amazon AWS or Microsoft Azure), thereby enabling dynamic integration of computing and data platforms [13], through the “Comet coordination space” (CometSpace). As illustrated in Figure 1, link to an externally hosted system, such as Amazon AWS is achieved through the use of worker gateways. CometCloud also exposes multiple resources from different cloud environments (that are connected through CometCloud) and present them as single pool of resources. In Figure 1, we have considered a scenario where the LEMS algorithm is deployed three different cloud space and the entire architecture is linked together using CometCloud. In each cloud or cluster that is part of the federated architecture workers are defined that are capable to execute LEMS operations this is achieved by deploying LEMS algorithm in each worker. However, the architecture is not limited to only three cloud space connected together it can be extended by connecting multiple clouds using the CometCloud programming model. The rationale to use multiple cloud in a federated environment over here is to handle increased computational requirement without affecting the execution time and address issues arising from traditional energy management system.

In CometCloud a master creates a task *tuple* which is stored in CometSpace, specifying where information about the task can be found, where the result are to be sent and which functions need to be executed for this task. Once the task is created and published in CometSpace, it can be picked up by a worker which can directly execute the task or act as a gateway to a third party system (e.g. Microsoft Azure) to enable task execution to be carried out. A task in this instance corresponds to a data analysis algorithm used to develop a predictive model. Use of multiple workers enables concurrent scenarios to be executed, thereby allowing what-if investigations to be carried out by a building manager, leading to the development of a number of different models (one per worker). In this case, each scenario models different arrival and departure times for electric vehicles at the building premises, and therefore the likelihood of utilizing the battery of such vehicles for energy storage. The actual number of workers to use in CometCloud is dependent on: (i) the potential variability within each forecast model; (ii) the diversity and rate of change of data collected, and (iii) the availability of computing resources to host worker tasks. Each worker can execute a task (representing a unique scenario in our case) in parallel without blocking resources or stopping regular operation of the local energy management system. Multiple workers can also be used to simulate various demand response scenarios to see the potential effect of these on building operation, cost of use and scheduling decisions for supporting load scheduling.

3. Simulation Scenario

Profile	Number of EV used	Avg. Arrival Time	Avg. Departure Time	Avg. Battery C	Avg. SOC on Connection	V2B	Nominal Power	Charger Efficiency
Profile 1	10	7:45	19:00	27KWh	84%	15%	3kW	0.95
Profile 2	10	8:45	19:00	27KWh	84%	15%	3kW	0.95
Profile 3	10	9:45	19:00	27KWh	84%	15%	3kW	0.95
Profile 4	10	10:45	19:00	27KWh	84%	15%	3kW	0.95
Profile 5	10	6:45	19:00	27KWh	84%	15%	3kW	0.95
Profile 6	10	11:45	19:00	27KWh	84%	15%	3kW	0.95

Table 1 EV assumptions

In order to demonstrate the scalability and computational speed of local energy management system we deployed LEMS on cloud using the comet cloud framework and we ran 12 simulation in parallel. The aim of each simulation was to see effect of dynamic factors such as EV arrival time, departure time, state of charge etc. on the total building demand. Furthermore, it would also identify the most desirable scenario that will be most effective in reducing the

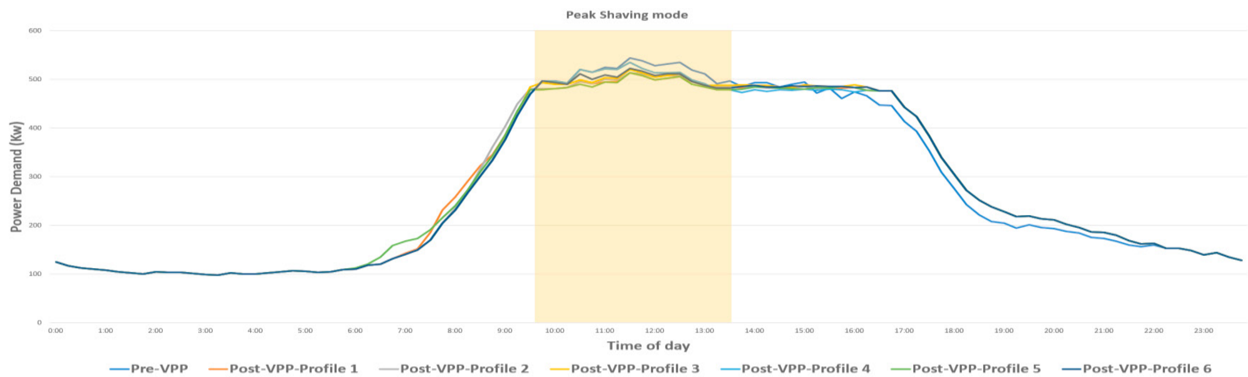

Figure 2 Power Demand profile of building using peak shaving operation at 15-minute time interval

total energy demand of a building. In order to identify the most desirable scenario we created six EV profiles and kept all attributes constant except the average EV arrival time. For each profile, a fleet of 10 electric vehicles were considered with a normal distribution of departure time and the average state of charge (SOC) on connection, battery capacity and other details are outlined in Table 1. It was also assumed that an EV driver was willing to discharge 15% of their vehicles battery to supply energy to the building (based on reduced cost charging incentive, for instance).
Based on these profiles we concurrently ran both operations (peak shaving and cost reduction modes) for each profile. Each task (created by the Master node on CometCloud) contained information about the operation being considered,

the location to find supporting data, and finally the location to store results. The master created 12 task and pushed them in the comet space, each task is picked up by available worker which later executes the task, compute the result and sends the result to the master. The output for each task is sent to the location specified in the task tuple. The LEMS algorithm is deployed at each worker across the federated architecture, this enables each worker to carry out any of the LEMS operation. The role of the master is to create the task tuple, keep track of all pending tasks and forward the computed charging/discharging schedule. The number of simulation are decided by the building manager based on the graphical user interface. In this manner, a building manager can create ‘n’ number of simulation and execute each of them by using the master- worker architecture. The building manager can use local energy management system to forecast building demand based on changing weather condition and then run these simulations to see the change in total building demand based on changing weather and EV attributes (arrival, departure etc.) Each simulation is executed within one time step and for our simulation a time step is defined as 15 minutes. This enables the building manager to see the effect of dynamically changing environment (EV delay in arrival, early departure, change in weather attributes, etc.) on demand profile of a commercial building by creating various simulation scenarios. Furthermore, based on the number of simulation scenarios the local energy management algorithm can be scaled up or down by adding/removing more master/workers containing local energy management algorithm in federated comet cloud.

Figure 2 shows changes in the demand profile of a building if we change the average EV arrival time while keeping all other parameter values constant. Based on simulations of the six profiles in Table 1, profile 5 (in which the average EV arrival time is 6:45 am) gives a maximum peak reduction of 5.5% and flattens the peak starting from

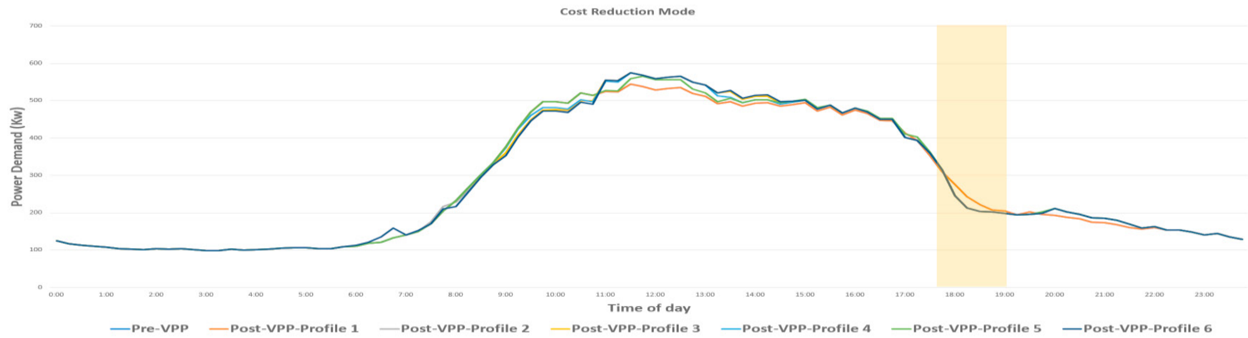

Figure 3 Power Demand profile of building using cost reduction operation at 15-minute time interval

9:30am and ending at 1:30pm by 4.5%. Similarly, figure 3 represents the demand profile of the building when cost reduction operation is run at 15 minute intervals. While running this algorithm the assumption was made that the triad peak can occur between 17:30 to 19:00 and accordingly a schedule was prepared in such a way that the total building demand was to be reduced during this period. We ran six simulation using the six profiles we had created with varying EV arrival time and we found that by using profile 5 in which the average EV arrival time was 6:45 am one can save a maximum of 8.1%. All the 12 simulation were executed parallelly in one time step that is under 15 minutes. Similarly, the building manager can execute multiple scenarios for a single or multiple (group of) building(s).

4. Conclusion

We describe the operation of a local energy management system that is capable of flattening the total peak demand of a building, reducing the total demand during most probable triad hours and enable a building manager to increase or decrease the total energy demand by creating a charging/discharging schedule for all connected electric vehicles. The local energy management system made use of the CometCloud framework that enables a building manager to scale up the algorithm to simultaneously carry out parallel simulation to identify the most cost effective configuration. The framework also enables the analysis algorithm being used to scale up easily when multiple buildings are considered. In the proposed system, a Master node can create a task tuple based on the configuration of each building and tasks can subsequently be executed by any available Worker (using locally available computational resources, or externally hosted services, such as through Amazon AWS). The building manager can create multiple scenarios using the graphical user interface, which is subsequently used to dynamically generate tasks for each simulation. The proposed infrastructure can scale dynamically based on the data size being generated, and the requirement of completing model development within a particular time deadline.

Acknowledgements

The authors would like to acknowledge the EPSRC project "Ebbs and Flows of Energy Systems" (EP/M507131/1) for supporting this work.

References

[1] M. Erol-Kantarci and H. T. Mouftah, "TOU-Aware Energy Management and Wireless Sensor Networks for Reducing Peak Load in Smart Grids," *2010 IEEE 72nd Veh. Technol. Conf. Fall, VTC2010-Fall*, pp. 1–5, 2010.

[2] U S Department of Energy, "Benefits of Demand Response in Electricity Markets and Recommendations for Achieving Them," *U.S. Dep. Energy*, no. February, p. 122, 2006.

[3] H. Kim, Y.-J. Kim, K. Yang, and M. Thottan, "Cloud-based demand response for smart grid: Architecture and distributed algorithms," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, 2011, pp. 398–403.

[4] X. Li and J.-C. Lo, "Pricing and peak aware scheduling algorithm for cloud computing," in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, 2012, pp. 1–7.

[5] C.-T. Yang, W.-S. Chen, K.-L. Huang, J.-C. Liu, W.-H. Hsu, and C.-H. Hsu, "Implementation of smart power management and service system on cloud computing," in *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on*, 2012, pp. 924–929.

[6] T. Rajeev and S. Ashok, "A cloud computing approach for power management of microgrids," in *Innovative Smart Grid Technologies-India (ISGT India), 2011 IEEE PES*, 2011, pp. 49–52.

[7] Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna, "Cloud-based software platform for big data analytics in smart grids," *Comput. Sci. Eng.*, vol. 15, no. 4, pp. 38–47, 2013.

[8] L. Tang, J. Li, and R. Wu, "Synergistic model of power system cloud computing based on Mobile-agent," in *Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on*, 2012, pp. 222–226.

[9] A. Javed, O. Rana, C. Marmaras, and L. Cipcigan, "Fog Paradigm for Local Energy Management Systems," in *2nd EAI International Conference on Cloud, Networking for IoT Systems*, 2017, pp. 1–10.

[10] C. Marmaras, A. Javed, L. Cipcigan, and O. Rana, "Predicting the energy demand of buildings during triad peaks in GB," *Energy Build.*, vol. 141, pp. 262–273, 2017.

[11] C. Marmaras, A. Javed, O. Rana, and L. M. Cipcigan, "A Cloud-Based Energy Management System for Building Managers," in *8th ACM/SPEC International Conference on Performance Engineering*, 2017.

[12] H. Kim and M. Parashar, "CometCloud: An autonomic cloud engine," *Cloud Comput. Princ. Paradig.*, pp. 275–297, 2011.

[13] J. Diaz-Montes, M. AbdelBaky, M. Zou, and M. Parashar, "Cometcloud: Enabling software-defined federations for end-to-end application workflows," *IEEE Internet Comput.*, vol. 19, no. 1, pp. 69–73, 2015.